Noise Simulation-Based Deep Optical Watermarking

Feng Wang[1], Hang Zhou[2], Han Fang[3], Weiming Zhang[1](✉), and Nenghai Yu[1]

[1] University of Science and Technology of China, Hefei, China
zhangwm@ustc.edu.cn
[2] Simon Fraser University, Vancouver, Canada
[3] National University of Singapore, Singapore, Singapore

Abstract. Digital watermarking is an important branch of information hiding, which effectively guarantees the robustness of embedded watermarks in distorted channels. To embed the watermark into the host carrier, traditional watermarking schemes often require the modification of the carrier. However, in some cases, the modification of the carrier is not allowed such as paintings in museums. To address such limitation, we utilize optical watermarking to embed the watermark into the host carrier. Optical watermarking refers to a technique that encodes the watermark into the visible light irradiating the object, where the watermark can be further extracted by the camera photography process. To realize transparency and robustness of the watermark, we propose a color-decomposition-based watermarking pattern generation algorithm which satisfies human visual system (HVS) characteristics, a camera shooting simulation algorithm which accurately produces the dataset for training, and a decoding network which can realize loss-less decoding of the embedded watermark. Various experiments demonstrate the superiority of our method and reveal the broad applicability of the proposed technique.

Keywords: Noise simulation · Optical watermarking · Human visual system

1 Introduction

Information hiding [1] refers to the technique of hiding secret information in the publicly available media so that people cannot be aware of its existence. As an important branch of information hiding, digital watermarking [2–4] can serve as a way to protect copyright or realize information transmission. The most important property of watermarking schemes is robustness, which directly influences the protection ability and transmission accuracy. To realize robustness, traditional schemes often embed the watermark into the stable coefficients of the carrier [5–8].

Although it is possible to achieve sufficient robustness with little perturbation, in some cases, even slight disturbance to the carrier is not allowed. For example, any damage is prohibited for the paintings displayed in the museum. Since paintings cannot be converted into electronic signals and cannot be modified, traditional watermarking

X. Sun et al. (Eds.): ICAIS 2022, LNCS 13340, pp. 283–298, 2022.
https://doi.org/10.1007/978-3-031-06791-4_23

techniques fail to be applied to the case. To address such limitation, we utilize the optical watermarking techniques [9–16] which can effectively realize the content-independent embedding. Optical watermarking refers to a technique that encodes the watermark signal into the visible light and projects the light onto the real object. With such a process, the object is unnecessary to be modified. At the extraction side, we use a camera to capture the irradiated object and decode the watermark by some image processing operations. Therefore, the content-independence and robustness can be both achieved.

Fig. 1. Visual fusing. Both images containing the pattern are fused to a clean image by human eyes.

Previous optical watermarking schemes often utilize the well-designed pattern to represent the watermark signal. The pattern should contain two important properties: transparency and robustness. Transparency refers to the visual quality after projecting the pattern onto the object, and robustness represents the extraction accuracy of captured images. However, there is an inherent contradiction between the two properties. So, how to guarantee robustness and visual quality at the same time is an unsolved problem worthy of further exploration.

To better balance transparency and robustness, we propose a novel noise simulation-based deep optical watermarking scheme. For transparency, we carefully study the human visual system (HVS) characteristics and propose a color-decomposition-based [16–19] watermarking pattern generation algorithm. Generally, it is based on the observation that human eyes will fuse two images into one if the two images are refreshed in a high frequency (no less than 60 Hz). Therefore, by alternatively projecting two complementary watermarked frames, human eyes can only see the synthetic frame. As shown in Fig. 1, some circular blocks are neatly arranged in the both images on the left. But human eyes would see the third image (HVS fuses the both images and composes the third image). Unlike HVS, the shutter speed of modern cameras is much higher and instead captures the decomposed frame that contains the pattern, making robustness possible. Moreover, we design a deep neural network at the extraction side to decode the watermark from the captured image. Given the assumption that the object surface is a flat 2D image, we generate the training dataset by simulating the projecting-shooting process, which achieves an approximate mapping from the generated pattern and the carrier image to the captured image.

In summary, our main contributions are three-fold:

- We propose a novel deep optical watermarking system that not only ensures the high visual quality but also realizes the strong robustness.
- To improve the extraction accuracy, we propose a generic noise-aware channel simulation model for the projecting-shooting process to create effective training data.
- Various experiment results demonstrate the superiority of our method compared with baseline methods and reveal broad application prospects of the proposed technique.

2 Related Work

2.1 Optical Watermarking

A series of works of optical watermarking have been presented [10–16] in the past few years, which can be divided into two categories: spatial-based methods and temporal-based methods.

For the first category, Uehira et al. [10] proposed to employ brightness-modulated light to embed invisible watermarking into objects. In [11], orthogonal transforms such Walsh-Hadamard Transform (WHT) and Discrete Cosine Transform (DCT) are utilized to generate the watermarking pattern. Uehira et al. [12] proposed the color difference-based modulation to represent the watermark and embed messages into the color difference signal Cb of Luminance, Chroma-blue and Chroma-red (YCbCr) signal to resist JPEG compression. However, the generated watermark patterns with these methods are often with poor visual quality and are obvious in human eyes.

As for the temporal-based methods, Unno et al. [13–16] proposed to introduce time modulation for better invisibility. In these schemes, two complementary watermarked images are generated and alternately displayed on a projector with a sufficient frequency. Although transparency is better, the message must be extracted via the video. Therefore, when facing the one-photo-capturing extraction, the watermarking scheme cannot be applied.

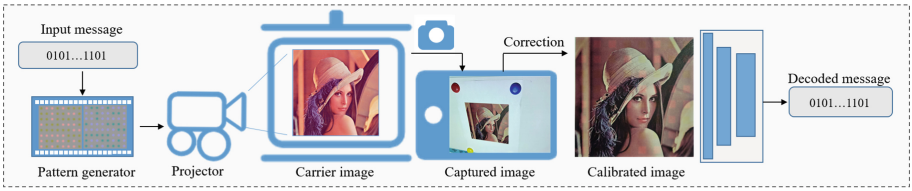

Fig. 2. The framework of optical watermarking system. First, the message is encoded into two patterns modulated positively and negatively. Second, the pattern scream is displayed temporally by the projector on the real object at 60 Hz. Third, a user takes a photo that contains the optical watermarking. Fourth, the captured image is transformed into a canonical image after perspective transformation. Finally, the calibrated image is fed into the following extracting network and the message hidden in the image is extracted.

2.2 Visual Illumination Model

Human Vision System. Human eyes can perceive changes in external light intensity such as flicker over time, but they cannot perceive flickering beyond a certain frequency, which is called flicker fusion threshold. The lowest frequency that causes flicker fusion is dubbed critical flicker frequency (CFF). CFF is generally considered to be about 60 Hz under most circumstances [20]. That is, when the flicker frequency is no less than 60 Hz, human eyes may not be able to observe the change. Besides, most projectors are designed to refresh at more than 60 Hz frequency to avoid visible flicker like screen devices [21]. Moreover, modern cameras can often capture the flicker since its shutter is shorter than the flicker cycle. With 60 Hz projectors, we can realize invisibility in eyes but recordable in camera.

Intrinsic Image Decomposition. The constituent elements of a natural image's appearance mainly include the illumination, shape and material [22]. As the Retinex theory shown in [23, 24], the image can be decomposed as the pixel-wise product of the illumination and the albedo, plus the specular component accounting for highlights due to viewpoint:

$$I(x, y) = S(x, y) \odot R(x, y) + C(x, y) \tag{1}$$

where $I(x, y)$ is the observed intensity at pixel (x, y), $S(x, y)$ is the illumination intensity, $R(x, y)$ is the albedo and $C(x, y)$ is the specular term. In the optical watermarking system, the captured image, the carrier image and the watermark pattern satisfy the aforementioned relationship. So based on the above equation, we could achieve a mapping from the carrier image and watermark to the captured image.

3 Method

In this section, we elaborate the proposed optical watermarking system. Figure 2 shows the basic framework, which consists of three parts: the pattern generator, the projector and the watermark extractor. The pattern generator is responsible for modulating the message into two complementary patterns. And the projector can alternatively project both patterns onto the carrier image. The final extractor can extract the watermark after correcting the image into the canonical image.

3.1 Pattern Generator

As a very important process, pattern generation determines the transparency of the watermarking. Projecting-shooting channel distortions are non-differential and the pattern is independent of the carrier, so we can't employ a deep learning-based method to generate an optimal pattern. Based on previous pattern generation algorithms [25, 26] and HVS, we propose a color-decomposition based watermarking pattern generation algorithm. For a message sequence of length L, we first reshape the sequence into a binary matrix with height h and width w (zeroing the part of $h \times w$). Based on the spatial arrangement

of messages, we employ a block of size $b \times b$ to represent 1-bit message so that the whole pattern size is $(b*h) \times (b*w)$. Formally, the 1-bit block can be generated by:

$$B(x,y) = \begin{cases} 1 - \left(\frac{D(x,y)}{\frac{b}{3}}\right)^2, & \text{if } D(x,y) \le b/3 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $D(x, y)$ indicates the distance between (x, y) and the center of the block:

$$D(x,y) = \sqrt{\left(x - \frac{b}{2}\right)^2 + \left(y - \frac{b}{2}\right)^2} \tag{3}$$

and (x, y) indicates the pixel coordinates of the image block. Considering the HVS is less sensitive to the red and blue components than the green component, we hide 1-bit message m into these two components and create two complementary templates $(+, -)$ for m:

$$B_{\pm}(r,g,b) = \begin{cases} [\beta \pm \alpha * B, \beta, \beta], & \text{if } m = 0 \\ [\beta, \beta, \beta \mp \alpha * B], & \text{otherwise} \end{cases} \tag{4}$$

where α controls the embedding intensity and $\alpha + \beta = 1$ for normalization. The generation of the whole pattern is by arranging each small block in the spatial order of the message matrix. After all the messages are embedded, we can generate two patterns, denoted by P_+ and P_-.

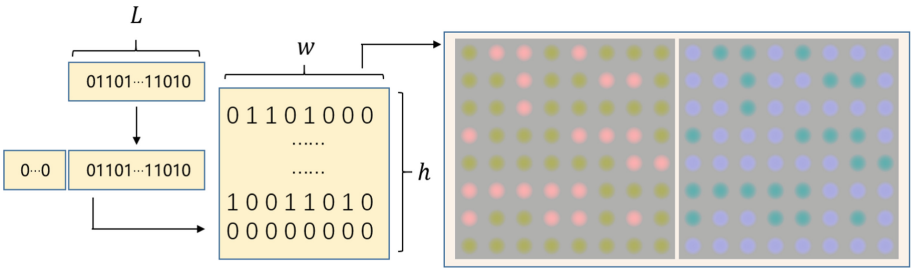

Fig. 3. Pattern generation. The sequence of bits of length L is resized to a binary matrix with height h and width w after zeroing the part of $w \times h - L$. Then the two patterns are modulated by the matrix positively and negatively.

3.2 Projector

The watermark embedding process is carried out by a projector, which alternatively projects the generated two complementary patterns with 60 Hz onto the object. Considering that the 60 Hz flicker is not perceptible to human eyes but recordable for cameras, both the transparency and the recorded ability can be satisfied. Practically, we often limit the projected pattern to tightly fit the carrier image for better performance.

3.3 Watermark Extractor

Perspective Correction. After captured, the captured image should be further perspectively corrected and fed into the decoding network. The correction process can be described as: after capturing the projected image by the camera, we detect the watermarking region and warp the region back to a rectangular image. We add a black border around the projected region and use the method in [27] to automatically locate 4 vertices of $V_1(x_1, y_1)$, $V_2(x_2, y_2)$, $V_3(x_3, y_3)$ and $V_4(x_4, y_4)$ as shown in Fig. 4. Then we set the transformation:

$$\begin{cases} x' = \frac{a_1 x + b_1 y + c_1}{a_0 x + b_0 y + 1}, \\ y' = \frac{a_2 x + b_2 y + c_2}{a_0 x + b_0 y + 1}. \end{cases} \tag{5}$$

(x', y') is the corresponding coordinate to these 4 vertices $V_1'\left(x_1', y_1'\right)$, $V_2'\left(x_2', y_2'\right)$, $V_3'\left(x_3', y_3'\right)$ and $V_4'\left(x_4', y_4'\right)$. Based on the equation, we can get 8 equations and solve them to obtain the value of eight variables. After that, we can form a stable mapping from the captured image to the calibrated image. Then the corrected image is cropped and resized to the input image size of the decoder network.

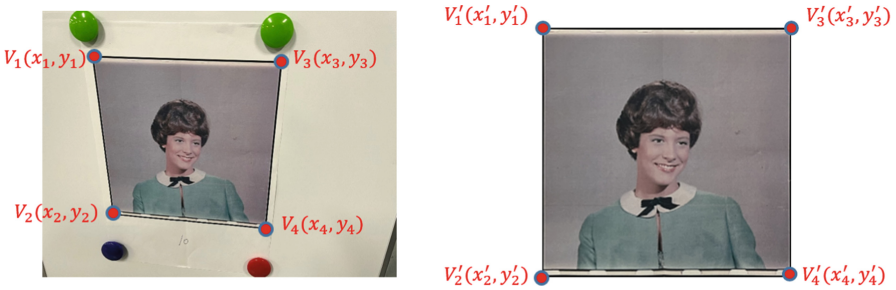

Fig. 4. Correction process. We add a black border around the image so we can automatically detect the marked four vertices. Then we can subsequently acquire the mapping from the captured image to the calibrated image. Note that the additional dots are magnets which are responsible for fixing the printed image to the board.

Simulated Dataset Generation. Since the final watermarking extraction is carried out by the decoding network, we need generate enough training data. Considering it is very complicated in time and effort to acquire real photo data, so instead, we propose an algorithm to simulate the distortions in the projecting-shooting process, as shown in Fig. 5. Assuming the projected object is a 2D plane with a wide variety of textures, the distortions in the actual physical process of projecting-shooting can be divided into three parts: The projecting color distortion, the synthesis distortion and the capturing distortion.

Projecting Color Distortion: Given the assumption that the projected pattern is spatially aligned with the carrier-image, we can approximate the projecting color distortions

by the pattern fusion and Gamma adjustment. Due to the shutter exposure effect, the captured image might consist of a part of the pattern P_+ and a part of pattern P_-, and the ratio is determined by the exposure time t_+ and t_-. Besides, the color of the projected P is quite different from its original status due to the hardware difference, so we utilize the Gamma adjustment with γ_1 to make an approximation. The whole simulation can be formulated by:

$$P(x, y) = \left[P_+(x, y) * t_+ + P_-(x, y) * t_-\right]^{\gamma_1} \tag{6}$$

where we set $t_+ + t_- = 1$ for normalization.

Synthesis Distortion: Since the embedding process is carried out by the projecting operation, the synthesis distortions mainly come from the lighting environment. Specifically, as mentioned above, the watermarking image is influenced by the projecting pattern $P(x, y)$, the carrier image $R(x, y$, the ambient illumination I_A and specular reflection I_C:

$$I(x, y) = (P(x, y) + I_A) \odot R(x, y) + I_C \tag{7}$$

where we assume I_A and I_C are constants over the entire image.

Capturing Distortion: Tancik et al. [28] proposed StegaStamp, which applied a set of differentiable image perturbations to simulate the print-shooting distortions during training. Similarly, we conclude the capturing process as four types of distortions: color manipulation, Gaussian noise, defocus blurring and JPEG compression.

As for color manipulation, there is an inherent difference between the captured image with its original color because of the sensor sampling operation. To better simulate such perturbations, we propose to utilize contrast adjustment, brightness shifting, hue shifting and Gamma adjustment on the watermarking image. We affine histogram rescaling $mx+n$ to achieve brightness shifting and contrast adjustment. For hue shifting, a random color offset s is added to each channel of RGB. Non-linear mismatching exists during shooting so we bring in Gamma adjustment with γ_2 to make an approximation.

In the camera shooting process, cameras may not fully focus on the target area, which will result in the defocus blurring distortion. To simulate the defocus blurring distortion, we perform a 5×5-sized Gaussian blurring operation on the image.

Due to the hardware components and the capturing environments, there are always different noises in the camera shooting process. Therefore, we directly employ Gaussian noise model with the standard deviation σ to represent the noise distortion that occurred during capturing.

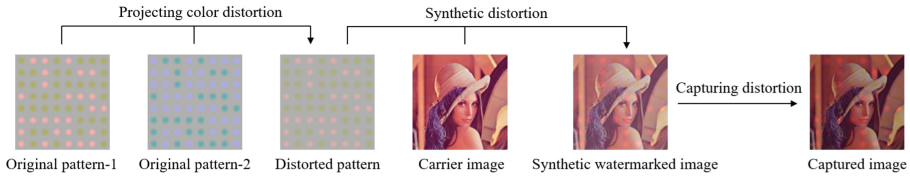

Fig. 5. Noise-aware channel simulation. First, original two patterns generate the distorted pattern under the influence of the projecting color distortion. Second, the synthetic watermarking image is synthesized by the distorted pattern and the carrier image. Finally, the captured image is generated from the synthetic watermarked image after capturing distortion.

JPEG compression distortion is introduced in the saving process since most cameras use JPEG as their default storage format. We simulate this process with JPEG compression of the quality factor Q.

Network Architecture. After generating the simulated dataset, we utilize them to train the decoding network. We employ two convolutional blocks (Conv-BatchNorm-ReLU) with a 3×3 kernel as the basic network unit and skip-connection [29] is used between neighboring units. When we use 32×32 pixels to represent 1 bit message, given the input image I, we first use five residual blocks with stride 1 and output features $F_1 \in \mathbb{R}^{64 \times H \times W}$. Then we use several down-sampled convolutional blocks to generate $F_2 \in \mathbb{R}^{256 \times \frac{H}{32} \times \frac{W}{32}}$. The last layer employs a convolutional layer of 1×1 and Tanh to generate the final output watermark $M' \in \mathbb{R}^{1 \times \frac{H}{32} \times \frac{W}{32}}$. During network training, we set l_1 distance as the message reconstruction loss, which can be formulated by:

$$L = \left|\left|M - M'\right|\right|_1 \tag{8}$$

where $M \in \{0, 1\}^{1 \times \frac{H}{32} \times \frac{W}{32}}$ is the original message matrix.

Fig. 6. Experimental system. The left side shows the whiteboard and the test carrier used for the environment, and the projector is placed on the table on the right. The additional magnets (colourful dots for fixing the carrier) are only needed for the sake of the experiment.

Table 1. Based on the real-world environment, we set up the scopes of these parameters. During training, we uniformly sample these parameters and generate simulated dataset in each step.

t_+	γ_1	I_A	I_C	m
$(0, 0.2) \cup (0.8, 1)$	$(1, 2.5)$	$(0.05, 0.1)$	$(0.15, 0.3)$	$(0.9, 1.1)$
n	s	γ_2	σ	Q
$(-0.2, 0.2)$	$(-0.1, 0.1)$	$(0.4, 1)$	$(0, 0.1)$	$(50, 100)$

Training Details. The decoder network is executed on NVIDIA GeForce RTX 2080Ti. For gradient descent, Adam [30] is applied with a learning rate of 10^{-3}. During training,

Table 2. Configuration parameters of conducting real-world experiments.

Device	Mobile phone	iPhone8 Plus
	Projector	EPSON
Position	Shooting distance	60 cm
	Projecting distance	50 cm
	Projecting angle	up 10°
Watermark	Embedding intensity	$\alpha = 30/255$
	Capacity	8× 8 bits
	Size	18.5 cm × 18.5 cm

we initialize the embedding intensity at $\alpha = 80/255$ and gradually decrease it to $30/255$. The decoder network is trained for 500 epochs with batch size 16. For training data, we randomly select 2595 images from ImageNet [31] and resize them to 256 × 256. For configuration parameters of the simulated dataset generation process, we uniformly sample them from scopes as shown in Table 1.

4 Real-World Experiments and Analysis

In this section, we first introduce the implementation details for our real-world experiments. Then comparative experiments of our method and baseline methods [10–12] are implemented. Finally, we implement additional experiments of the proposed method.

4.1 Implementation Details

The test dataset is USC-SIPI image dataset with 14 images [32]. All the test images are printed in A4 paper with 300dpi. The default projector and mobile phone we used are "EPSON EB-C301MN" with the refresh rate of 60 Hz and "iPhone8 Plus" with the camera resolution of 4032 × 3024. We use the embedding intensity $\alpha = 30/255$ for test. The details of the default experimental configuration are shown in Table 2, and our experimental system is shown in Fig. 6. To measure the visual quality of the watermarking image, we perform a mean opinion score (MOS) test. We ask 25 persons to assign a score from 1 (bad quality, the watermark clearly visible) to 5 (excellent quality, the watermark invisible) at the default shooting position. We use the mean value of every observer's score to represent the final MOS of the scheme. The robustness is evaluated by the average extraction accuracy of the captured images. We compare the messages and calculate the bit accuracy rate (Fig. 7).

Fig. 7. Visual qualities. We can see gray blocks in images with baseline methods.

Table 3. Mean opinion score (MOS) test compared with baseline methods.

Method	DCT [9]	DWT [10]	DCT-Cb [11]	Proposed
MOS	2.094	1.844	2.781	**4.144**

4.2 Visual Quality Comparison

The visual quality comparison results (MOS) are shown in Table 3. We can easily find that the proposed method achieves better MOS than other baseline methods. As shown in Fig. 3, the original images and captured images with different methods are displayed. In human eyes, the proposed projected pattern is just a pure white pattern with a certain intensity. But for the other schemes, obvious texture can be easily perceived. This is because the proposed method leverages the insensitivity of HVS with flicker, which greatly improves the visual quality of the watermarked image.

4.3 Robustness Tests

We conduct experiments on different conditions to compare the extraction performance of our method with baseline methods [10–12]. Explicitly speaking, the captured distances range from 30 cm to 90 cm and the captured angles range from left 75° to right 75°. As shown in Table 4, the proposed method maintains the best extraction bit accuracy in most distance cases, except for 30 cm and 90 cm. We can see that at 40 cm–80 cm, the proposed scheme can achieve the accuracy beyond 88%, and the closer shooting distance will result in better performance, which can be analyzed that closer shooting distance could get a clearer photograph, indicating higher bit accuracy. However, we can find that the proposed method doesn't perform best at 30 cm in all distance cases. We analyze the

reason as: our method is based on color difference in different channels and shooting too closely can cause the captured image too bright, causing the color difference signal to disappear. And other methods are based on frequency modulation, so they are supposed to achieve the best performance with the clearest picture at the closest distance. Table 5 shows the performances at different angles. It can be easily seen that our method could achieve more than 90% of bit accuracy within angle [−15°,15°], and even at an angle of left 75°, the accuracy could still reach 71.63%. The performance on the front is not always best in all angle cases. That may be attributed to more intense specular reflection in the frontal.

Table 4. Bit accuracy (%) comparison of extracted message with different shooting distances.

Distance/cm	DCT [9]	DWT [10]	DCT-Cb [11]	Proposed
30 cm	91.74	92.35	**93.97**	90.85
40 cm	88.39	88.62	91.07	**96.88**
50 cm	90.40	92.13	92.63	**94.42**
60 cm	84.49	78.52	74.00	**90.07**
70 cm	80.58	82.31	85.38	**92.75**
80 cm	84.49	78.18	82.92	**88.84**
90 cm	86.38	87.05	**89.84**	83.37

Table 5. Bit accuracy (%) comparison of extracted message with different shooting angles.

Angle	DCT [9]	DWT [10]	DCT-Cb [11]	Proposed
Left 75°	60.94	63.39	61.61	**71.63**
Left 60°	80.47	81.58	**82.92**	77.79
Left 45°	78.01	77.46	82.59	**89.40**
Left 30°	83.15	82.03	82.37	**85.38**
Left 15°	77.57	76.45	80.92	**93.53**
Frontal	84.49	78.52	74.00	**90.07**
Right 15°	82.70	79.24	84.93	**92.97**
Right 30°	82.03	80.13	85.49	**95.54**
Right 45°	78.35	79.07	80.92	**87.72**
Right 60°	78.91	78.13	77.34	**89.17**
Right 75°	55.52	57.25	55.47	**86.38**

4.4 Additional Experiments

The Influence of the Noise-Aware Channel Simulation. Since the network is trained with the simulated data, the simulation performance greatly influences the network performance. In this section, we mainly explore the importance of different simulating operations with the following cases. The bit accuracy (%) results are shown in Tables 6 and 7, column (1): Without projecting color distortion (randomly selecting one from two patterns); column (2): Without synthesis distortion (regarding carrier images as whiteboards); column (3): Without capturing distortion; column (4): Iden (including all distortions). In all distortions, it's not hard to find that synthetic distortion is the most important among the three distortions. That's because the Retinex theory-based synthesis explains the basic process of the captured image generation.

Table 6. The influence of the noise-aware channel simulation with different distances.

Distance	W/o projecting color distortion	W/o synthesis distortion	W/o capturing distortion	Iden
30 cm	87.17	74.55	86.83	**90.85**
40 cm	95.76	83.48	95.42	**96.88**
50 cm	92.75	82.14	90.96	**94.42**
60 cm	86.61	75.56	85.16	**90.07**
70 cm	87.95	75.11	86.72	**92.75**
80 cm	82.81	73.77	83.26	**88.84**
90 cm	80.02	69.31	79.69	**83.37**

The Influence of Embedding Intensity. Embedding intensity α significantly influences the extraction accuracy in the real-world test. In this section, we mainly show and discuss the influence of embedding intensity. To determine appropriate α, we conduct a test on different intensities from 10/255 to 50/255 with the step of 10/255. For each intensity, we conduct the MOS test and the robustness test with the default setting. The corresponding results are shown in Table 8. It can be found that as the intensity increases, the visual quality gradually decreases while the robustness gradually increases. The reason is that although human eyes are not sensitive when flickering with 60 Hz, frequency is not the only restriction. When the intensity achieves a certain value, such artifacts can still be found even. Therefore, we should take a careful trade-off of visual quality and robustness and select the appropriate intensity $\alpha = 30/255$.

Adaptability to Different Devices. To reveals the versatility of the proposed method on different devices, we use five mobile phones ("iPhone8 Plus", "Mix2S", "Mi4", "Honor V20", "iPhone SE") and two projectors ("EPSON EB-C301MN", "NEC CR3117X") to test the extraction accuracy at the distance of 60 cm from the frontal. As shown in Table 9, the extraction accuracy is beyond 82% in all device pairs, which indicates the

Table 7. The influence of the noise-aware channel simulation with different angles.

Angle	W/o projecting color distortion	W/o synthesis distortion	W/o capturing distortion	Iden
L 75°	64.62	61.94	64.06	**71.63**
L 60°	73.32	64.51	72.21	**77.79**
L 45°	81.81	71.21	82.14	**89.40**
L 30°	79.80	67.30	79.80	**85.38**
L 15°	91.29	76.56	90.96	**93.53**
Frontal	86.61	75.56	85.16	**90.07**
R 15°	89.40	76.56	87.83	**92.97**
R 30°	94.31	77.12	92.08	**95.54**
R 45°	85.60	70.98	81.81	**87.72**
R 60°	86.05	72.32	84.49	**89.17**
R 75°	77.46	68.19	76.56	**86.38**

versatility of the proposed method. Besides, we found that the extraction accuracy with "NEC CR3117X" is higher than that with "EPSON EB-C301MN". We conclude that the "NEC CR3117X" has a higher projection resolution which influences the capture accuracy at the camera side.

Table 8. MOS-Accuracy. The performance across a range of intensity.

α(1/255)	10	20	30	40	50
MOS	4.631	4.378	4.144	3.063	2.563
Accuracy (%)	70.65	77.57	90.07	94.08	94.75

Table 9. Bit accuracy (%) on different mobile phones and projectors.

Device	iPhone8 Plus	Mi 4	Mix2S	Honor V20	iPhone SE
EPSON	90.07	88.73	86.72	84.49	82.59
NEC	90.4	92.75	88.84	96.65	95.09

The Influence of Different 1-bit Block Sizes. In this section, we main explore the Influence of different block sizes that represent 1-bit message. To better clarify The Influence of different sizes, we utilize the size ranging from 8×8 to 64×64 pixels

Table 10. Bit accuracy (%) comparison with different 1-bit block sizes.

Block size	64 × 64	32 × 32	16 × 16	8 × 8
Accuracy	94.64	90.07	87.77	73.96

to represent 1-bit message (32 × 32 default) and adaptively change the stride for the network to re-train the network. Then we test these cases to generate the results shown in Table 10. We can easily find that the extraction accuracy increases when the block size is larger. We conclude that: when the block size is smaller, more possible distortions are introduced after capturing, so the extraction accuracy is poorer.

5 Conclusion

In this paper, we introduce a novel optical watermarking scheme based on noise simulation and deep neural network. To achieve better transparency, we utilize color-decomposition to embed watermark. As for robustness, we propose the noise-aware channel simulation model to generate the training dataset and employ the decoder network to extract the message. Extensive experiments demonstrate the superiority compared with baseline methods and reveal the broad applicability of our method.

Funding Statement. This work was supported in part by the Natural Science Foundation of China under Grant 62072421, 62002334, 62102386, 62121002 and U20B2047, Anhui Science Foundation of China under Grant 2008085QF296, Exploration Fund Project of University of Science and Technology of China under Grant YD3480002001, and by Fundamental Research Funds for the Central Universities WK5290000001.

References

1. Petitcolas, F.A., Anderson, R.J., Kuhn, M.G.: Information hiding-a survey. Proc. IEEE **87**(7), 1062–1078 (1999)
2. Katzenbeisser, S., Petitcolas, F.A.P.: Digital Watermarking, vol. 2. Springer, Artech House, London (2000)
3. Xiong, L., Han, X., Yang, C., Shi, Y.Q.: Robust reversible watermarking in encrypted image with secure multi-party based on lightweight cryptography. IEEE Trans. Circ. Syst. Video Technol. **32**, 75–91 (2021). https://doi.org/10.1109/TCSVT.2021.3055072
4. Bhaskar, A., Sharma, C., Mohiuddin, K., Singh, A., Nasr, O.A.: A robust video watermarking scheme with squirrel search algorithm. Comput. Mater. Continua **71**(2), 3069–3089 (2022)
5. Pereira, S., Pun, T.: Robust template matching for affine resistant image watermarks. IEEE Trans. Image Process. **9**(6), 1123–1129 (2000)
6. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process. **6**(12), 1673–1687 (1997)
7. Alhumyani, H., Alrube, I., Alsharif, S., Afifi, A., Amar, C.B.: Analytic beta-wavelet transform-based digital image watermarking for secure transmission. Comput. Mater. Continua **70**(3), 4657–4673 (2022)

8. Fang, H., Zhang, W., Zhou, H., Cui, H., Yu, N.: Screen-shooting resilient watermarking. IEEE Trans. Inf. Forensics Secur. **14**(6), 1403–1418 (2018)
9. Uehira, K., Suzuki, K., Ikeda, H.: Applications of optoelectronic watermarking technique to new business and industry systems utilizing flat-panel displays and smart devices. In: IEEE Industry Application Society Annual Meeting, pp. 1–9 (2014)
10. Uehira, K., Suzuki, M.: Digital watermarking technique using brightness-modulated light. In: IEEE International Conference on Multimedia and Expo, pp. 257–260 (2008)
11. Ishikawa, Y., Uehira, K., Yanaka, K.: Practical evaluation of illumination watermarking technique using orthogonal transforms. J. Display Technol. **6**(9), 351–358 (2010)
12. Uehira, K., Unno, H.: Effects of JPEG compression on reading optical watermarking embedded by using color-difference modulation. J. Comput. Commun. **6**(1), 56–64 (2017)
13. Oshita, K., Unno, H., Uehira, K.: Optically written watermarking technology using temporally and spatially luminance-modulated light. Electron. Imaging **2016**(8), 1–6 (2016)
14. Unno, H., Uehira, K.: Lighting technique for attaching invisible information onto real objects using temporally and spatially color-intensity modulated light. IEEE Trans. Ind. Appl. **56**(6), 7202–7207 (2020)
15. Unno, H., Yamkum, R., Bunporn, C., Uehira, K.: A new displaying technology for information hiding using temporally brightness modulated pattern. IEEE Trans. Ind. Appl. **53**(1), 596–601 (2016)
16. Unno, H., Uehira, K.: Display technique for embedding information in real object images using temporally and spatially luminance-modulated light. IEEE Trans. Ind. Appl. **53**(6), 5966–5971 (2017)
17. Cui, H., Bian, H., Zhang, W., Yu, N.: Unseencode: invisible on-screen barcode with image-based extraction. In: IEEE Conference on Computer Communications, pp. 1315–1323 (2019)
18. Zhang, L., et al.: Kaleido: you can watch it but cannot record it. In: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, pp. 372–385 (2015)
19. Song, K., Liu, N., Gao, Z., Zhang, J., Zhai, G., Zhang, X.P.: Deep restoration of invisible QR code from TPVM display. In: 2020 IEEE International Conference on Multimedia & Expo Workshops, pp. 1–6 (2020)
20. Nomura, Y., et al.: Evaluation of critical flicker-fusion frequency measurement methods using a touchscreen-based visual temporal discrimination task in the behaving mouse. Neurosci. Res. **148**, 28–33 (2019)
21. Menozzi, M., Lang, F., Naepflin, U., Zeller, C., Krueger, H.: CRT versus LCD: effects of refresh rate, display technology and background luminance in visual performance. Displays **22**(3), 79–85 (2001)
22. Liu, Y., Li, Y., You, S., Lu, F.: Unsupervised learning for intrinsic image decomposition from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3248–3257 (2020)
23. Land, E.H., Mccann, J.J.: Lightness and retinex theory. Josa **61**(1), 1–11 (1971)
24. Grosse, R., Johnson, M.K., Adelson, E.H., Freeman, W.T.: Ground truth dataset and baseline evaluations for intrinsic image algorithms. In: Proceedings of the IEEE Conference on Computer Vision, pp. 2335–2342 (2009)
25. Gugelmann, D., Sommer, D., Lenders, V., Happe, M., Vanbever, L.: Screen watermarking for data theft investigation and attribution. In: 10th International Conference on Cyber Conflict, pp. 391–408 (2018)
26. Fang, H., et al.: Deep template-based watermarking. IEEE Trans. Circ. Syst. Video Technol. **31**(4), 1436–1451 (2020)
27. Katayama, A., Nakamura, T., Yamamuro, M., Sonehara, N.: New high-speed frame detection method: Side trace algorithm (STA) for i-appli on cellular phones to detect watermarks. In: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, pp. 109–116 (2004)

28. Tancik, M., Mildenhall, B., Ng, R.: Stegastamp: invisible hyperlinks in physical photographs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2117–2126 (2020)
29. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
30. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (2014)
31. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F.: ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
32. The USC-SIPI image database (2020). http://sipi.usc.edu/database/